US 20140040478A1

(19) **United States**

(12) **Patent Application Publication**

Hsu et al.

(10) **Pub. No.: US 2014/0040478 A1**

(43) **Pub. Date: Feb. 6, 2014**

(54) **GLOBAL SERVER LOAD BALANCING**

(71) Applicant: **Brocade Communications Systems, Inc.**, San Jose, CA (US)

(72) Inventors: **Ivy Pei-Shan Hsu**, Pleasanton, CA (US); **David Chun-Ying Cheung**, Cupertino, CA (US); **Rajkumar Ramniranjan Jalan**, Saratoga, CA (US)

(21) Appl. No.: **13/925,670**

(22) Filed: **Jun. 24, 2013**

Related U.S. Application Data

(63) Continuation of application No. 12/496,560, filed on Jul. 1, 2009, now Pat. No. 8,504,721, which is a continuation of application No. 11/741,480, filed on Apr. 27, 2007, now Pat. No. 7,581,009, which is a continuation of application No. 09/670,487, filed on Sep. 26, 2000, now Pat. No. 7,454,500.

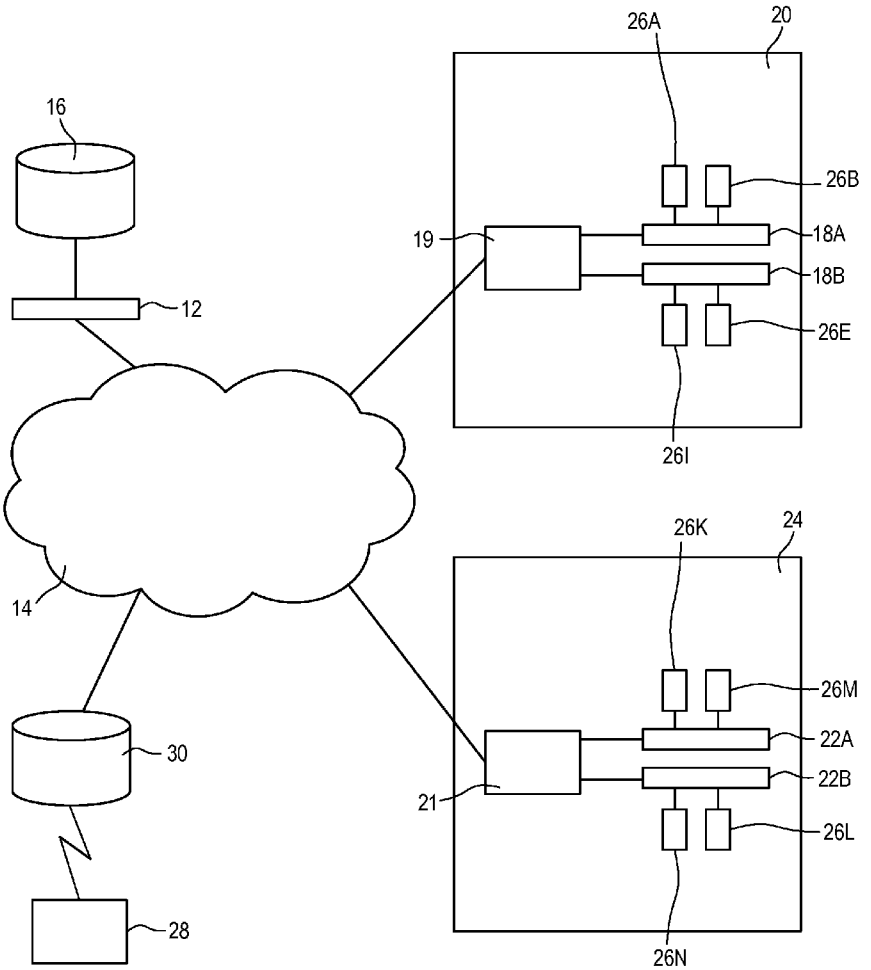
Publication Classification

(51) **Int. Cl.**
H04L 29/08 (2006.01)

(52) **U.S. Cl.**
CPC *H04L 67/1002* (2013.01)
USPC **709/226**

(57) **ABSTRACT**

A global server load balancing (GSLB) switch serves as a proxy to an authoritative DNS communicates with numerous site switches which are coupled to host servers serving specific applications. The GSLB switch receives from site switches operational information regarding host servers within the site switches neighborhood. When a client program requests a resolution of a host name, the GSLB switch, acting as a proxy of an authoritative DNS, returns one or more ordered IP addresses for the host name. The IP addresses are ordered using metrics that include the information collected from the site switches. In one instance, the GSLB switch places the address that is deemed "best" at the top of the list.



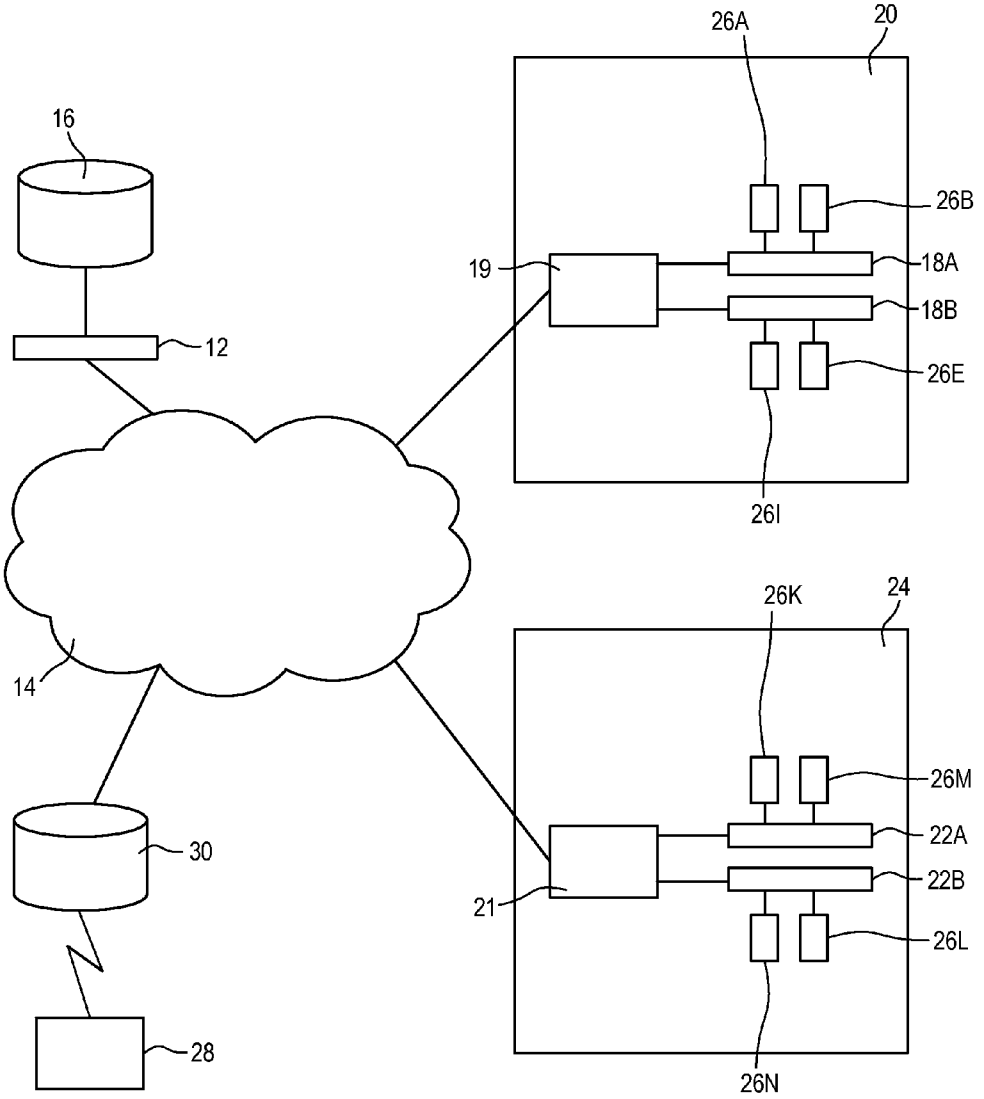

26A
20
16
26B
19
18A
18B
12
26E
26I
26K
24
14
26M
22A
30
22B
21
26L
28
26N

FIG. 1

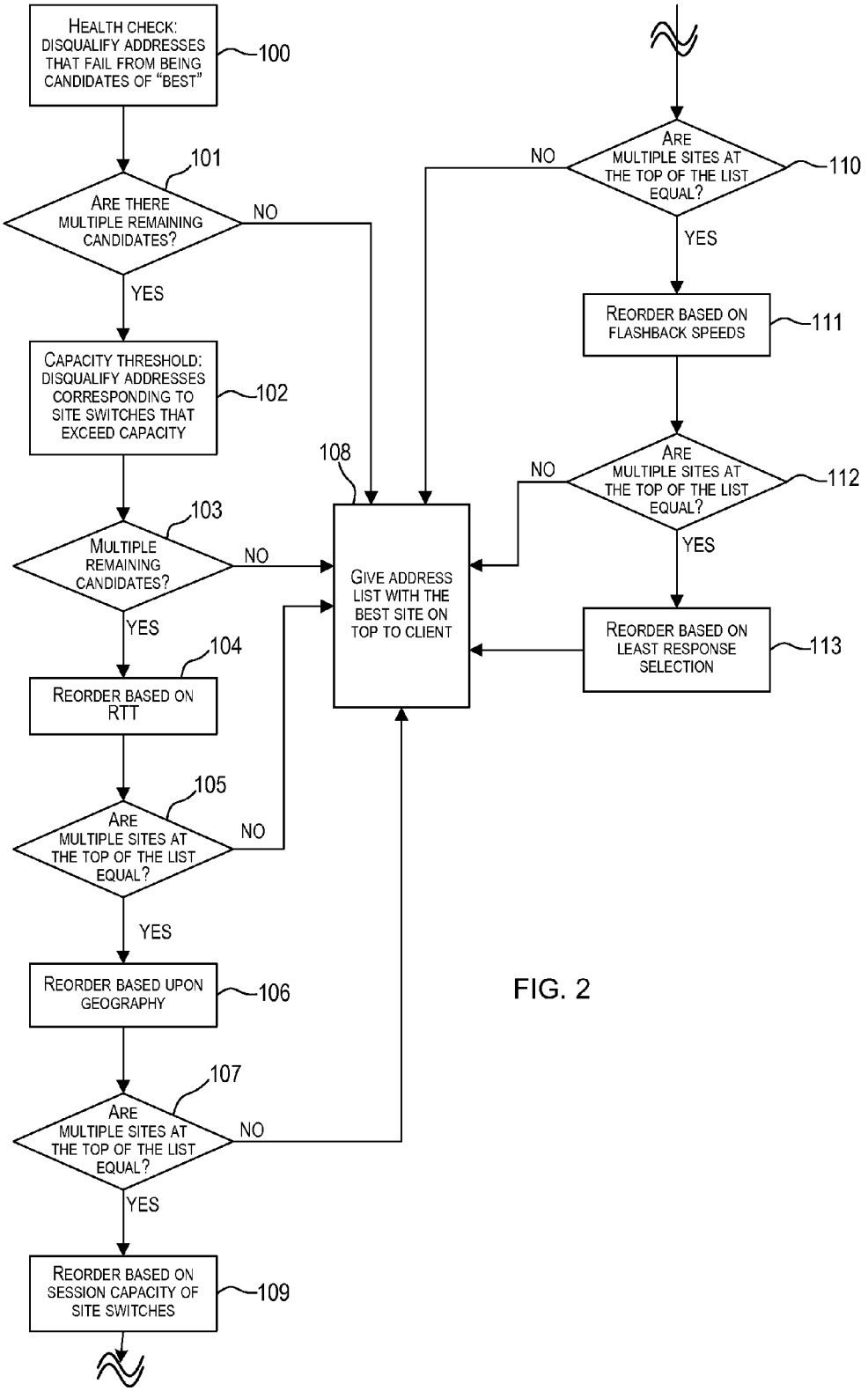

HEALTH CHECK: DISQUALIFY ADDRESSES THAT FAIL FROM BEING CANDIDATES OF "BEST"
100
ARE THERE MULTIPLE REMAINING CANDIDATES?
101
NO
YES
CAPACITY THRESHOLD: DISQUALIFY ADDRESSES CORRESPONDING TO SITE SWITCHES THAT EXCEED CAPACITY
102
MULTIPLE REMAINING CANDIDATES?
103
NO
YES
REORDER BASED ON RTT
104
ARE MULTIPLE SITES AT THE TOP OF THE LIST EQUAL?
105
NO
YES
REORDER BASED UPON GEOGRAPHY
106
ARE MULTIPLE SITES AT THE TOP OF THE LIST EQUAL?
107
NO
YES
REORDER BASED ON SESSION CAPACITY OF SITE SWITCHES
109
108
GIVE ADDRESS LIST WITH THE BEST SITE ON TOP TO CLIENT
ARE MULTIPLE SITES AT THE TOP OF THE LIST EQUAL?
110
NO
YES
REORDER BASED ON FLASHBACK SPEEDS
111
ARE MULTIPLE SITES AT THE TOP OF THE LIST EQUAL?
112
NO
YES
REORDER BASED ON LEAST RESPONSE SELECTION
113

FIG. 2

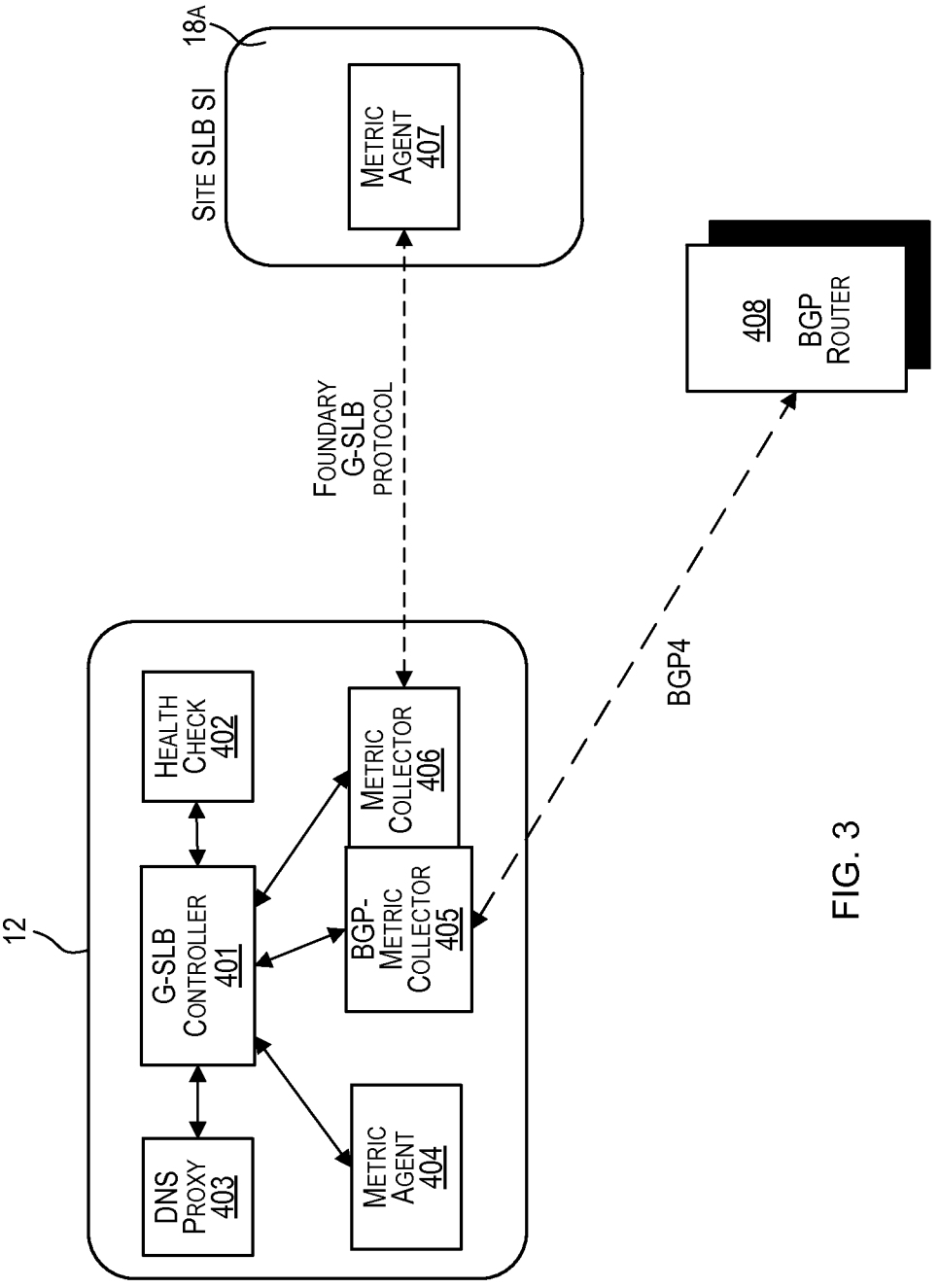

12
G-SLB CONTROLLER 401
HEALTH CHECK 402
DNS PROXY 403
METRIC AGENT 404
BGP-METRIC COLLECTOR 405
METRIC COLLECTOR 406
FOUNDARY G-SLB PROTOCOL
SITE SLB SI
18A
METRIC AGENT 407
BGP4
408 BGP ROUTER

FIG. 3

GLOBAL SERVER LOAD BALANCING

CROSS-REFERENCES TO RELATED APPLICATIONS

[0001] The present application is a continuation of U.S. application Ser. No. 12/496,560, filed Jul. 1, 2009 titled Global Server Load Balancing, which is a continuation of U.S. application Ser. No. 11/741,480, filed Apr. 27, 2007 titled Global Server Load Balancing, now U.S. Pat. No. 7,581,009 issued Aug. 25, 2009, which is a continuation of U.S. application Ser. No. 09/670,487 filed Sep. 26, 2000 titled Global Server Load Balancing, now U.S. Pat. No. 7,454,500 issued Nov. 18, 2008. The entire contents of the aforementioned applications are herein incorporated by reference in their entirety for all purposes.

BACKGROUND

[0002] 1. Field of the Invention

[0003] The present invention relates to load balancing among servers. More particularly, the present invention relates to achieving load balancing by, in response to resolving a DNS query by a client, providing the address of a server that is expected to serve the client with a high performance in a given application.

[0004] 2. Description of the Related Art

[0005] Under the TCP/IP protocol, when a client provides a symbolic name ("URL") to request access to an application program or another type of resource, the host name portion of the URL needs to be resolved into an IP address of a server for that application program or resource. For example, the URL (e.g., http://www.foundrynet.com/index.htm) includes a host name portion www.foundrynet.com that needs to be resolved into an IP address. The host name portion is first provided by the client to a local name resolver, which then queries a local DNS server to obtain a corresponding IP address. If a corresponding IP address is not locally cached at the time of the query, or if the "time-to-live" (TTL) of a corresponding IP address cached locally has expired, the DNS server then acts as a resolver and dispatches a recursive query to another DNS server. This process is repeated until an authoritative DNS server for the domain (i.e. foundrynet.com, in this example) is reached. The authoritative DNS server returns one or more IP addresses, each corresponding to an address at which a server hosting the application ("host server") under the host name can be reached. These IP addresses are propagated back via the local DNS server to the original resolver. The application at the client then uses one of the IP addresses to establish a TCP connection with the corresponding host server. Each DNS server caches the list of IP addresses received from the authoritative DNS for responding to future queries regarding the same host name, until the TTL of the IP addresses expires.

[0006] To provide some load sharing among the host servers, many authoritative DNS servers use a simple round-robin algorithm to rotate the IP addresses in a list of responsive IP addresses, so as to distribute equally the requests for access among the host servers.

[0007] The conventional method described above for resolving a host name to its IP addresses has several shortcomings. First, the authoritative DNS does not detect a server that is down. Consequently, the authoritative DNS server continues to return a disabled host server's IP address until an external agent updates the authoritative DNS server's resource records. Second, when providing its list of IP addresses, the authoritative DNS sever does not take into consideration the host servers' locations relative to the client. The geographical distance between the server and a client is a factor affecting the response time for the client's access to the host server. For example, traffic conditions being equal, a client from Japan could receive better response time from a host server in Japan than from a host server in New York. Further, the conventional DNS algorithm allows invalid IP addresses (e.g., that corresponding to a downed server) to persist in a local DNS server until the TTL for the invalid IP address expires.

BRIEF SUMMARY

[0008] The present invention provides an improved method and system for serving IP addresses to a client, based on a selected set of performance metrics. In accordance with this invention, a global server load-balancing (GSLB) switch is provided as a proxy for an authoritative DNS server, together with one or more site switches each associated with one or more host servers. Both the GSLB switch and the site switch can be implemented using the same type of switch hardware. Each site switch provides the GSLB switch with current site-specific information regarding the host servers associated with the site switch. Under the present invention, when an authoritative DNS server resolves a host name in a query and returns one or more IP addresses, the GSLB switch filters the IP addresses using the performance metrics compiled from the site-specific information collected from the site switches. The GSLB switch then returns a ranked or weighted list of IP addresses to the inquirer. In one embodiment, the IP address that is estimated to provide the best expected performance for the client is placed at the top of the list. Examples of suitable performance metrics include availability metrics (e.g., a server's or an application's health), load metrics (e.g., a site switch's session capacity or a corresponding preset threshold), and proximity metrics (e.g., a round-trip time between the site switch and a requesting DNS server, the geographic location of the host server, the topological distance between the host server and the client program). (A topological distance is the number of hops between the server and the client). Another proximity metrics is the site switch's "flashback" speed (i.e., how quickly a switch receives a health check result). The ordered list can also be governed by other policies, such as the least selected host server.

[0009] The present invention is better understood upon consideration of the detailed description of the preferred embodiments below, in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] FIG. **1** illustrates a global server load balancing configuration.

[0011] FIG. **2** illustrates in a flow chart one embodiment of the algorithm for selecting the "best" address from the list of addresses supplied by the authoritative DNS.

[0012] FIG. **3** is a block diagram showing the functional modules of GSLB switch **12** and site switch **18***a* relevant to the global server load balancing function.

DETAILED DESCRIPTION

[0013] FIG. **1** illustrates one embodiment of the present invention that provides a global server load balancing configuration. As shown in FIG. **1**, global server load balancing

(GSLB) switch **12** is connected to Internet **14** and acts as a proxy to an authoritative Domain Name System (DNS) server **16** for the domain "foundrynet.com" (for example). That is, while the actual DNS service is provided by DNS server **16**, the IP address known to the rest of the Internet for the authoritative DNS sever of the domain "foundrynet.com" is a virtual IP address configured on GSLB switch **12**. Of course, DNS server **16** can also act simultaneously as an authoritative DNS for other domains. GSLB switch **12** communicates, via Internet **14**, with site switches **18**A and **18**B at site **20**, site switches **22**A and **22**B at site **24**, and any other similarly configured site switches. Site switch **18**A, **18**B, **22**A and **22**B are shown, for example, connected to routers **19** and **21** respectively and to servers **26**A, . . . , **26**I, . . . **26**N. Some or all of servers **26**A, . . . , **26**I, . . . , **26**N may host application server programs (e.g., http and ftp) relevant to the present invention. These host servers are reached through site switches **18**A, **18**B, **22**A and **22**B using one or more virtual IP addresses configured at the site switches, which act as proxies to the host servers. A suitable switch for implementing either GSLB switch **12** or any of site switches **18**A, **18**B, **22**A and **22**B is the "ServerIron" product available from Foundry Networks, Inc.

[0014] FIG. **1** also shows client program **28** connected to Internet **14**, and communicates with local DNS server **30**. When a browser on client **28** requests a web page, for example, using a Universal Resource Locator (URL), such as http://www.foundrynet.com/index.htm, a query is sent to local DNS server **30** to resolve the symbolic host name www.foundrynet.com to an IP address of a host server. The client program receives from DNS server **30** a list of IP addresses corresponding to the resolved host name. This list of IP addresses is either retrieved from local DNS server **30**'s cache, if the TTL of the responsive IP addresses in the cache has not expired, or obtained from GSLB switch **12**, as a result of a recursive query. Unlike the prior art, however, this list of IP addresses are ordered by GSLB switch **12** based on performance metrics described in further detail below. In the remainder of this detailed description, for the purpose of illustrating the present invention only, the list of IP addresses returned are assumed to be the virtual IP addresses configured on the proxy servers at switches **18**A, **18**B, **22**A and **22**B (sites **20** and **24**). In one embodiment, GSLB switch **12** determines which site switch would provide the best expected performance (e.g., response time) for client **28** and returns the IP address list with a virtual IP address configured at that site switch placed at the top. (Within the scope of the present invention, other forms of ranking or weighting the IP addresses in the list can also be possible.) Client program **28** can receive the ordered list of IP addresses, and typically selects the first IP address on the list to access the corresponding host server.

[0015] FIG. **3** is a block diagram showing the functional modules of GSLB switch **12** and site switch **18***a* relevant to the global server load balancing function. As shown in FIG. **3**, GSLB **12** includes a GSLB switch controller **401**, health check module **402**, DNS proxy module **403**, metric agent **404**, routing metric collector **405**, and site-specific metric collector **406**. GSLB switch controller **401** provides general control functions for the operation of GSLB switch **12**. Health check module **402** is responsible for querying, either periodically or on demand, host servers and relevant applications hosted on the host servers to determine the "health" (i.e., whether or not it is available) of each host server and each relevant application. Site-specific metric collector **406** communicates with metric agents in site-specific switches (e.g., FIG. **3** shows site-specific metric collector **406** communicating with site-specific metric agent **407**) to collect site-specific metrics (e.g., number of available sessions on a specific host server). Similarly, routing metric collector **405** collects routing information from routers (e.g., topological distances between nodes on the Internet). FIG. **3** shows, for example, router **408** providing routing metric collector **405** with routing metrics (e.g., topological distance between the load balancing switch and the router), using the Border Gateway Protocol (BGP). DNS proxy module **403** (a) receives incoming DNS requests, (b) provides the host names to be resolved to DNS server **16**, (c) receives from DNS server **16** a list of responsive IP addresses, (d) orders the IP addresses on the list received from DNS server **16** according to the present invention, using the metrics collected by routing-metric collector **405** and site specific collector **406**, and values of any other relevant parameter, and (e) provides the ordered list of IP addresses to the requesting DNS server. Since GSLB switch **12** can also act as a site switch, GSLB switch **12** is provided site-specific metric agent **404** for collecting metrics for a site-specific metric collector.

[0016] In one embodiment, the metrics used in a GSLB switch includes (a) the health of each host server and selected applications, (b) each site switch's session capacity threshold, (c) the round trip time (RTT) between a site switch and a client in a previous access, (d) the geographical location of a host server, (e) the current available session capacity in each site switch, (f) the "flashback" speed between each site switch and the GSLB switch (i.e., how quickly each site switch responds to a health check from the GSLB switch), and (g) a policy called the "Least Response selection" (LRS) which prefers the site least selected previously. Many of these performance metrics can be provided default values. Each individual metric can be used in any order and each metric can be disabled. In one embodiment, the LRS metric is always enabled.

[0017] FIG. **2** illustrates in a flow diagram one embodiment of an optimization algorithm utilized by GSLB switch **12** to process the IP address list received from DNS server **16**, in response to a query resulting from client program **28**. As shown in FIG. **2**, in act **100**, upon receiving the IP address list from DNS server **16**, GSLB switch **12** performs, for each IP address on the IP address list (e.g., host server **26**I connected to site switch **18**B), a layer **4** health check and a layer **7** check. Here, layers **4** and **7** refer respectively to the transport and application protocols in the Open System Interconnection (OSI) protocol layers. The layer **4** health check can be a Transmission Control Protocol (TCP) health check or a User Datagram Protocol (UDP) health check. Such a health check can be achieved, for example, by a "ping-like" operation defined under the relevant protocol. For example, under the TCP protocol, a TCP SYN packet can be sent, and the health of the target is established when a corresponding TCP ACK packet is received back from the target. In this embodiment, the layer **7** health check is provided for specified applications, such as the well-known HyperText Transport Protocol (HTTP) and the File Transfer Protocol (FTP) applications. If a host server or an associated application fails any of the health checks it is disqualified (act **102**) from being the "best" site and may be excluded from the IP address list to be returned to client program **28**. Since the health check indicates whether or not a host server or an associated application is available, the health check metric is suitable for use to eliminate an IP address from the candidates for the "best" IP

address (i.e., the host server expected to provide the highest performance). After act **100**, if the list of IP addresses consists of only one IP address (act **101**), the list of IP addresses is returned to client program **28** at act **108**.

[0018] After act **100**, if the list of candidate IP addresses for the best site consists of multiple IP addresses, it is further assessed in act **102** based upon the capacity threshold of the site switch serving that IP address. Each site switch may have a different maximum number of TCP sessions it can serve. For example, the default number for the "ServerIron" product of Foundry Network is one million sessions, although it can be configured to a lower number. The virtual IP address configured at site switch **18**B may be disqualified from being the "best" IP address if the number of sessions for switch **18**B exceed a predetermined threshold percentage (e.g., 90%) of the maximum number of sessions. (Of course, the threshold value of 90% of the maximum capacity can be changed.) After act **102**, if the list of IP addresses consists of only one IP address (act **103**), the list of IP addresses is returned to client program **28** at list **108**.

[0019] After act **102**, if the IP address list consists of multiple IP addresses (act **103**), the remaining IP addresses on the list can then be reordered in act **104** based upon a round-trip time (RTT) between the site switch for the IP address (e.g., site switch **18**B) and the client (e.g., client **28**). The RTT is computed for the interval between the time when a client machine requests a TCP connection to a proxy server configured on a site switch, sending the proxy server a TCP SYN packet, and the time a site switch receives from the client program a TCP ACK packet. (In response to the TCP SYN packet, a host server sends a TCP SYN ACK packet, to indicate acceptance of a TCP connection; the client machine returns a TCP ACK packet to complete the setting up of the TCP connection.) The GSLB Switch (e.g., GSLB switch **12**) maintains a database of RTT, which it creates and updates from data received periodically from the site switches (e.g., site switches **18**A, **18**B, **22**A and **22**B). Each site collects and stores RTT data for each TCP connection established with a client machine. In one embodiment, the GSLB switch favors one host server over another only if the difference in their RTTs with a client machine is greater than a specified percentage, the default specified percentage value being 10%. To prevent bias, the GSLB) switch ignores, by default, RTT values for 5% of client queries from each responding network. After act **105**, if the top entries on the list of IP addresses do not have equal RTTs, the list of IP addresses is returned to client program **28** at act **108**.

[0020] If multiple sites have equal RTTs then the list is reordered in act **106** based upon the location (geography) of the host server. The geographic location of a server is determined according to whether the IP address is a real address or a virtual IP address ("VIP"). For a real IP address the geographical region for the host server can be determined from the IP address itself. Under IANA, regional registries RIPE (Europe), APNIC (Asia/Pacific Rim) and ARIN (the Americas and Africa) are each assigned different prefix blocks. In one embodiment, an IP address administered by one of these regional registries is assumed to correspond to a machine located inside the geographical area administered by the regional registry. For a VIP, the geographic region is determined from the management IP address of the corresponding site switch. Of course, a geographical region can be prescribed for any IP address to override the geographic region determined from the procedure above. The GSLB Switch prefers an IP address that is in the same geographical region as the client machine. At act **107**, if the top two entries on the IP list are not equally ranked, the IP list is sent to the client program **28** at act **108**.

[0021] After act **106**, if multiple sites are of equal rank for the best site, the IP addresses can then be reordered based upon available session capacity (act **109**). For example, if switch **18**A has 1,000,000 sessions available and switch **22**B has 800,000 sessions available, switch **18**A is then be preferred, if a tolerance limit, representing the difference in sessions available expressed as a percentage of capacity in the larger switch, is exceeded. For example, if the tolerance limit is 10%, switch **18**A will have to have at a minimum 100,000 more sessions available than switch **22**B to be preferred. If an IP address is preferred (act **110**), the IP address will be placed at the top of the IP address list, and is then returned to the requesting entity at act **108**. Otherwise, if the session capacity does not resolve the best IP address, act **111** then attempts to a resolution based upon a "flashback" speed. The flashback speed is a time required for a site switch to respond to layers **4** and **7** health checks by the GSLB switch. The flashback speed is thus a measure of the load on the host server. Again, the preferred IP address will correspond to a flashback speed exceeding the next one by a preset tolerance limit.

[0022] In one embodiment, flashback speeds are measured for well-known applications (layer **7**) and their corresponding TCP ports (layer **4**). For other applications, flashback speeds are measured for user selected TCP ports. Layer **7** (application-level) flashback speeds are compared first, if applicable. If the application flashbacks fail to provide a best IP address, layer **4** flashback speeds are compared. If a host server is associated with multiple applications, the GSLB switch selects the slowest response time among the applications for the comparison. At act **112**, if a best IP address is resolved, the IP address list is sent to client program **28** at act **108**. Otherwise, at act **113**, an IP address in the site that is least often selected to be the "best" site is chosen. The IP address list is then sent to client program **28** (act **108**).

[0023] Upon receipt of the IP address list, the client's program uses the best IP address selected (i.e., the top of the list) to establish a TCP connection with a host server. Even then, if there is a sudden traffic surge that causes a host server to be overloaded, or if the host servers or the applications at the site become unavailable in the mean time, the site switch can redirect the TCP connection request to another IP address using, for example, an existing HTTP redirection procedure. The present invention does not prevent a site switch from performing load balancing among host servers within its subnetwork by redirection using a similar mechanism.

[0024] To provide an RTT under the present invention described above, at the first time a client accesses an IP address, a site switch (e.g., site switch **22**A of FIG. **2**) monitors the RTT time—the time difference between receiving a TCP SYN and a TCP ACK for the TCP connection—and records it in an entry of the cache database. The RTT time measured this way corresponds to the natural traffic flow between the client machine and the host sever specified, rather than an artificial RTT based on "pinging" the client machine under a standard network protocol. Periodically, the site switches report the RTT database to a GSLB switch along with load conditions (e.g., number of sessions available). The GSLB switch aggregates the RTTs reported into a proximity table indexed by network neighborhood. (A network neighborhood is the portion of a network sharing a prefix of an IP

address.) The GSLB switch can thus look up the RTT for a client machine to any specific host server, based on the client's network neighborhood specified in the client's IP address. From the accesses to the host servers from a large number of network neighborhoods, the GSLB switch can build a comprehensive proximity knowledge database that enables smarter site selection. In order to keep the proximity table useful and up-to-date, the GSLB switch manages the proximity table with cache management policies (e.g., purging infrequently used entries in favor of recently obtained RTTs). The proximity data can be used for all IP addresses served by each site switch.

[0025] While particular embodiments of the present invention have been shown and described it will be apparent to those skilled in the art that changes and modifications may be made without departing from this invention in its broader aspect and, therefore, the appended claims are to encompass within their scope all such changes and modifications.

1. A method comprising:

receiving, by a network device from a domain name server, a plurality of network addresses generated by the domain name server in response to a domain name query originated by a client machine;

determining, by the network device, based upon a first metric associated with a plurality of sites, that a single network address from the plurality of network addresses is not better than all other network addresses in the plurality of network addresses for responding to the domain name query;

responsive to the determining, processing, by the network device, one or more network addresses from the plurality of network addresses using a second metric associated with the plurality of sites, wherein the second set metric is different from the first metric; and

causing, by the network device, a list of multiple network addresses from the plurality of network addresses to be forwarded to the client machine.

2. The method of claim **1** wherein the first metric is round trip times associated with the plurality of sites, wherein the round trip time associated with a site is indicative of time for exchanging a message between a switch at the site and the client machine.

3. The method of claim **1** further comprising:

determining, by the network device, that the processing using the second metric does not yield a single network address as better than other network addresses in the plurality of network addresses for responding to the domain name query;

processing, by the network device, one or more network addresses from the plurality of network addresses using a third metric associated with the plurality of sites, wherein the third metric is different from the first metric and the second metric.

4. The method of claim **1** wherein:

processing the one or more network addresses from the plurality of network addresses using the second metric comprises generating an ordered list of multiple network addresses from the plurality of network addresses;

the causing comprises causing the ordered list of multiple network addresses from the plurality of network addresses to be forwarded to the client machine.

5. The method of claim **1** wherein the plurality of network addresses comprises a first virtual IP address configured at a host server site switch at a first site from the plurality of sites, the host server site switch associated with one or more host servers at the first site, the one or more host servers being reachable via the host server site switch, the method further comprising:

receiving, by the network device, information related to the first metric information and information related to the second metric for the first network address from the host server site switch; and

wherein the determining comprises processing the plurality of network addresses based upon the first metric comprises using the information related to the first metric;

wherein processing the one or more network addresses using the second metric. comprises using the information related to the second metric.

6. A network device comprising:

a memory; and

processor;

wherein the network device is configurable to:

receive, from a domain name server, a plurality of network addresses generated by the domain name server in response to a domain name query originated by a client machine;

determine, based upon a first metric associated with a plurality of sites, that a single network address from the plurality of network addresses is not better than all other network addresses in the plurality of network addresses for responding to the domain name query;

process one or more network addresses from the plurality of network addresses using a second metric associated with the plurality of sites, wherein the second set metric is different from the first metric; and

cause a list of multiple network addresses from the plurality of network addresses to be forwarded to the client machine.

7. The network device of claim **5** wherein the first metric is round trip times associated with the plurality of sites, wherein the round trip time associated with a site is indicative of time for exchanging a message between a switch at the site and the client machine.

8. The network device of claim **5** further configurable to:

determine that the processing using the second metric does not yield a single network address as better than other network addresses in the plurality of network addresses for responding to the domain name query; and

process one or more network addresses from the plurality of network addresses using a third metric associated with the plurality of sites, wherein the third metric is different from the first metric and the second metric.

9. The network device of claim **5** further configurable to:

generate an ordered list of multiple network addresses from the plurality of network addresses as a result of processing performed using the second metric; and

cause the ordered list of multiple network addresses to be forwarded to the client machine.

* * * * *